## Remarks/Arguments

With reference to the Office Action mailed May 18, 2007, Applicants offer the following remarks and argument.

### Status of the Claims

Claims 1-17 were originally presented for examination.

The claims were subject to a restriction requirement. Applicants elected claims 1-9 with traverse, and amended claims 1, 4, 6, and 8. Claim 1 was objected to (correction has been made). Claims 1-9 were rejected. Claims 1-5, 8, and 9 were rejected as being unpatentable over Tang in view of Kanno, and Gilmour. Claims 6 and 7 were rejected as being unpatentable Tang in view Gilmour and Kanno.

### The Art of Record

The primary reference, United States Patent 6,636,849 to Tang et al. for Data Search Employing Metric Spaces, Multi-grid Indexes, And B-grid Trees describes systems and methods for generating indexes and fast searching of "approximate", "fuzzy", or "homologous" matches for a large quantity of data in a metric space. The data is indexed to generate a search tree taxonomy. Once the index is generated, a query can be provided to report all hits within a certain neighborhood of the query. In an even faster implementation, Tang et al. describe using their disclosed method together with existing approximate sequence comparison algorithms, such as FASTA and BLAST. As described by Tang et al., a local distance of a local metric space is used to generate local search tree branches. This may include homology search for DNA and/or protein sequences, textual or byte-based searches, literature search based on lists of keywords, and vector and matrix based indexing and searching.

However, as will be described below, Tange et al. do not disclose Applicants' claimed invention.

United States Patent 7,007,019 to Kanno et al. for Vector Index Preparing Method, Similar Vector Searching Method, And Apparatuses For The Methods describes a method for searching a vector from a large, dimensional vector database using a single vector index, and using either (1) a measure of an inner product or (2) a distance, by designating a similarity search range and maximum obtained pieces number. Vector index preparation is performed by decomposing each vector into a plurality of partial vectors and characterizing the vector by (1) a norm division, (2) a belonging region, and (3) a declination division, to thereby prepare an index. Next, Kanno et al. describe similarity searching by (1) obtaining a partial query vector and partial search range from a query vector and search range, (2) performing similarity search in each partial space to accumulate a difference from the search range and to obtain an upper limit value, and (3) obtaining a correct measure from a higher upper limit value to obtain a final similarity search result.

The third reference is United States Patent 6,377,949 to Gilmour for Method And Apparatus For Assigning A Confidence Level To A Term Within A User Knowledge Profile. Gilmour describes a method of assigning a confidence level to a term within an electronic document, such as an e-mail. This includes the step of determining a quantitative indicator, for example, an occurrence value. This occurrence value is based on the number of occurrences of a particular term within an electronic document, and associating the occurrence term within the relevant term. Next, a qualitative indicator, based on a quality of the term, is determined. This qualitative indicator may be determined utilizing the parts of speech of words comprising the term. A confidence level value, which may be utilized to indicate a relative importance of the term in describing a user knowledge base, is generated utilizing the quantitative and qualitative indicators.

**The Office Action of May 18, 2007**

Art Rejections

In the Office Action of May 18, 2007, the claims (claims 1-9) were rejected as anticipated by Tang et al. with various combinations and permutations of Kanno and Gilmour.

**Discussion**

The overarching issue presented is whether Applicants' amendments impart allowability to the amended claims. Claim 1 (as amended) is typical:

> A computer system for generating data structures for information retrieval of documents stored in a database, said documents being stored as document-keyword vectors generated from a predetermined keyword list, and said document-keyword vectors forming nodes of a hierarchical structure imposed upon said documents, said computer system comprising:
>
> <u>a document-key word matrix generation subsystem;</u>
>
> a neighborhood patch generation subsystem for generating groups of nodes having similarities as determined using a search structure, said neighborhood patch generation subsystem including a subsystem for generating a <u>spatial approximation sample hierarchy</u> ~~hierarchical~~ structure upon said document-keyword vectors and a patch defining subsystem for creating patch relationships among said nodes with respect to a metric distance between nodes;
>
> a query vector generation subsystem accepting search conditions and query keywords, generating a corresponding query vector, and storing the generated query vector;
>
> [[a]] <u>an intra-patch confidence and intrapath</u> confidence determination subsystem <u>for every element of the database, the spatial approximation sample hierarchy structure computing a neighborhood patch consisting of a list of those database elements most similar to it</u> for computing intra-patch confidence values between patches and interpath confidence values; ~~and~~
>
> <u>a self confidence determining subsystem for (a) computing a list of self confidence values, for every stored patch, (b) computing relative self confidence values, and (c) thereafter using the relative self confidence values to determine a size of a best subset of each patch to serve as a cluster candidate;</u>
>
> a cluster estimation subsystem for generating cluster data of said document-keyword vectors using said similarities of patches wherein the cluster estimation subsystem selects said patches depending on ~~inner patch~~ <u>intra-patch</u> confidence values to represent clusters

of said document keyword vectors, estimate the sizes of said patches, and generate cluster data of document keyword vectors using similarities of the patches; and

a redundant cluster elimination subsystem for using the inner patch confidence values to eliminate redundant cluster candidates.

Tang, Kanno, and Gilmour have been applied to original claim 1 as follows:

| Claim 1 As Amended | References |
|---|---|
| A computer system for generating data structures for information retrieval of documents stored in a database, said documents being stored as document-keyword vectors generated from a predetermined keyword list, and said document-keyword vectors forming nodes of a hierarchical structure imposed upon said documents, said computer system comprising: | |
| a document-key word matrix generation subsystem; | **NEWLY ADDED CLAIM LIMITATION** |
| a neighborhood patch generation subsystem for generating groups of nodes having similarities as determined using a search structure, said neighborhood patch generation subsystem including a subsystem for generating a spatial approximation sample hierarchy ~~hierarchical~~ structure upon said document-keyword vectors and a patch defining subsystem for creating patch relationships among said nodes with respect to a metric distance between nodes; | **CONTAINS NEWLY ADDED CLAIM LIMITATIONS**<br><br>Tang, Column 4, lines 39-54:<br>In operation, the systems and methods of the present invention first process a data set to create a multigrid tree. The multigrid tree comprises gridpoints (e.g. a data element in the data space comprising of the data set or, stated differently, a collection of adjacent points in the data space). The multigrid tree is calculated using distance functions of a metric space. Associated with each grid point is a radius that defines the neighborhood of the grid point (i.e., a grid). In an illustrative implementation, the multigrid tree comprises a plurality of descending branches that originate from a root grid point. The further the branch from the root grid point, the smaller the radius of the grid points residing on that branch. The multigrid tree may be a Bgrid tree that is balanced such that data elements of the data set are partitioned in equal size grids such that search time is more homogenous for varying search queries.<br><br>Tang, Column 11, lines 18-27:<br>The grid concept can be extended one more step such that there are multiple levels of grids. Each grid at a fixed level can be subdivided into smaller grids with smaller radius (i.e. a smaller neighborhood). Those smaller grids become children, and the original grid with its gridpoint is the parent. In this way, multiple levels of grids can be linked via parentchild relationships. As illustrated in FIG. 5, the multilayered grid structure when assembled forms a grid search tree.<br><br>Tang, Column 10, line 61 – column 11, line 5:<br>As shown in FIG. 4, metric space "E" 400 can be divided into many small grids 405, 410, 415, etc., each containing a grid point, 405a, 410a, 415a, etc, respectively. FIG. 4 shows an example of a multigrid in a 2dimensional point set with L.sub.1 distance and a corresponding search performed on the grid. For example, consider a set of points E, all which are located in a 2dimensional area of [1,5][1,4]. The L.sub.1 distance may be defined as follows, given p.sub.1 =(x.sub.1,y.sub.1), p.sub.2 =(x.sub.2,y.sub.2), d(p.sub.1,p.sub.2)=max(.vertline.x.sub.1 x.sub.2.vertline.,.vertline.y.sub.1 y.sub.2.vertline.). Using this calculated distance, an exemplary search may be performed to answer the question: given a query point q=(2.2, 1.8), find out all points p within the area that satisfy d(q,p)<0.3.<br><br>Tang, Column 4, Line 55- Column 5, Line 17:<br>For example, for a given query point q, inexact matches to q in a |

| Claim 1 As Amended | References |
|---|---|
| | given data set can be found. In mathematical terms, the search aims to find all points p in the data set such that those points p satisfy d(q,p).ltoreq..epsilon., where d( . , . ) is the distance function in the metric space, and .epsilon. is the size of interested neighborhood. When a search started, a comparison is performed among all of the grid points of at the first level of the created Bgrid tree. At each level, many subtrees are totally eliminated for further search by applying the triangular inequality rule. For example, suppose the grid points are g.sub.ij, the comparison search for the desired data elements to satisfy the search string q is only to be further carried out within those grids where d(q,g.sub.ij)<.epsilon.+.delta..sub.ij, where .delta..sub.ij is the chosen grid size. The systems and methods of the present invention perform these calculations to produce result set for communication to the participating user.<br><br>In an alternative implementation, the systems and methods of the present invention are used to calculate local distances for submitted search queries. For example, for a given query q, the search aims to find most of points p where d(p, q)<.epsilon., whereas the missed points p are likely close to the boundary of d(p,q)=.epsilon.. In this implementation, current search algorithms, such as, BLAST and FASTA are used to create a local multigrid tree (or a local Bgrid tree) having local distances. Employing the same steps above, the local multigrid tree (or local Bgrid) tree is analyzed to find data elements for the submitted search query. Since local distances are used to create the local multigrid tree (or the local Bgrid search tree), the result set will contain most of the desired hits for a submitted search query.<br><br>Tang, Column 13, line 61 – Column 14, line 61:<br><br>The process of building the Bgrid tree of FIG. 6A is described by the flow diagram of FIG. 5. This method is a slightly modified approach compared with the Btree definitions described by R. Bayer and E. McCreight, "Organization and Maintenance of Large Ordered Indexes," Acta Informatica. 1:173189 (1970), which is herein incorporated by reference. The Bgrid tree of FIG. 6A maintains each subtree within its parent grid; whereas in the conventional Btree definition, the subtree is either to the left or right of its parent node. This slight difference makes the Bgrid tree concept uniform to all space dimensions in a metric space.<br><br>FIG. 8B shows an exemplary Bgrid tree of order 4 in a 2dimensioanl metric space. Similar to the Bgrid tree of FIG. 8A, each grid is defined by a grid point 850, a radius 855, and some descriptions 860. A shown, there are four grids g.sub.1 (865), g.sub.2 (870), g.sub.3 (875), and g.sub.4 (880). The neighborhoods defined by the grid points and the radii may be overlapping, but as the descriptions indicate, these neighborhoods exist separate and apart. Thus, the grids at the same level have no overlapping regions (points). The descriptions are provided to assign the points in overlapping regions to one of the grids. |
| a query vector generation subsystem accepting search conditions and query keywords, generating a corresponding query vector, and storing the generated query vector; | Tang, Column 4, line 55 to column 5, line 3:<br>For example, for a given query point q, inexact matches to q in a given data set can be found. In mathematical terms, the search aims to find all points p in the data set such that those points p satisfy d(q,p).ltoreq..epsilon., where d( . , . ) is the distance function in the metric space, and .epsilon. is the size of interested neighborhood. When a search started, a comparison is performed among all of the grid points of at the first level of the created Bgrid tree. At each level, many subtrees are totally eliminated for further search by applying the triangular inequality rule. For example, suppose the grid points are g.sub.ij, the comparison |

| Claim 1 As Amended | References |
|---|---|
| | search for the desired data elements to satisfy the search string q is only to be further carried out within those grids where d(q,g.sub.ij)<.epsilon.+.delta..sub.ij, where .delta..sub.ij is the chosen grid size. The systems and methods of the present invention perform these calculations to produce result set for communication to the participating user.<br><br>Tang, Column 7, lines 32-41:<br>The present invention may also be employed to perform keyword searches on large volumes of literature data. The literature data is transposed to a metric space such that the distance function is defined as linear function of shared keywords. In operation, a keyword is provided to the search system and method, using the newly defined distance function, the search will aim to find occurrences of the submitted keyword (or keywords) in the literature data set and report those literature data elements that have share occurrences.<br><br>Tang, Column 11, line 64 to column 12, line 32:<br><br>Once created, the multigrid tree can be searched to find exact or approximate or homologous matches for a search query. The multigrid tree can be searched to provide a solution to the following example. Suppose a multigrid search tree representing a set E in a metric space, and a query point in the metric space is provided. The task to find all the points p (exact matches) in E such that d(q,p)<=.epsilon.. If .epsilon.=0 may be accomplished by the following. FIG. 6 shows the processing performed to find "exact" or "inexact" matches within a multigrid search tree. The search routine starts at block 600 from the root grids. The search query is then obtained at block 610. The search then begins at level "l" having the list of grids (g.sub.il, . . . ,g.sub.ik) left to search at block 620. For each grid point along level "l", a check is then made at block 630 to ascertain all of the grid points of all the subtrees of (g.sub.il, . . . ,g.sub.ik). This check is realized by a comparison of the children grid points with the query. A decision is then performed at block 640 using the triangle inequality to discard any children that is no longer of interest (i.e. a check to see if the subtree grid point satisfies the equation d(g.sub.ij,q)>.delta..sub.ij +.epsilon.). If the analyzed grid point of the child (i.e. subtree) does not satisfy the inequality, the subtree is dropped from the search at block 650 and processing ends at block 690. If, however, the alternative proves to be true, processing proceeds from block 640 to block 660 where the subtree is kept as part of the search. Processing then proceeds to block 670 where a check is performed to determine if the currently analyzed level is the last level of the multigrid tree. If it is the last level, processing proceeds to block 680 where all of the matches for the search query are reported. Processing then terminates at block 690. If, however, the check at block 670 proves that the currently analyzed multigrid tree level is not the last level of the multigrid tree, processing reverts to block 630 and proceeds therefrom.<br><br>Kanno, Column 15, lines 34-40:<br>Partial query condition calculation means 303 calculates a partial inner product lower limit value f as a lower limit value of an inner product of 37 types of 8-dimensional partial query vectors q with the partial vector corresponding to q by f=.alpha.\|q\|.sup.2/\|Q\|.sup.2 with respect to partial spaces of 0 to 36 for the query vector Q obtained by the search condition input means 302.<br><br>Kanno, Column 5, lines 1-11:<br>FIG. 1 is a block diagram showing a whole constitution of the first embodiment of a vector index preparing apparatus according to claims 1, 3 to 8, 14, 16 to 21 of the present invention. In FIG. |

| Claim 1 As Amended | References |
|---|---|
| | 1, a vector database 101 stores 200,000 pieces of vector data constituted of two items of: a 296-dimensional unit real vector prepared from a newspaper article full text database of 200,000 collected newspaper articles and indicating characteristic of each newspaper article; and an identification number in a range of 1 to 200,000, and has a content as shown in FIGS. 12A and 12B. |
| [[a]] <u>an intra-patch confidence and intrapath</u> confidence determination subsystem <u>for every element of the database, the spatial approximation sample hierarchy structure computing a neighborhood patch consisting of a list of those database elements most similar to it</u> for computing intra-patch confidence values between patches and interpath confidence values; ~~and~~ | <u>**CONTAINS NEWLY ADDED CLAIM LIMITATIONS**</u><br><br>Tang, column 16, lines 25-39:<br><br>A local alignment can be transformed into a local distance function. For example, S.sub.a, S.sub.b, S.sub.c, having a local alignment shown in FIG. 9 is provided. It is assumed that in the overlapped region, the match is 100%. We define d.sub.a (S.sub.1,S.sub.2)=len(S.sub.1)+len(S.sub.2)2 len(S.sub.1 S.sub.2), where len( ) is the length of a sequence, and S.sub.1 S.sub.2 define the overlapped region. As such,<br><br>b)+len(S.sub.c)+2(len(S.sub.a)len(S.sub.a S.sub.b)len(S.sub.a S<br><br>c)).gtoreq.len(S.sub.b)+len(S.sub.c)=d.sub.a (S.sub.b,S.sub.c) since len(S a)len(S.sub.a S.sub.b)len(S.sub.a S.sub.c)>0. |
| <u>a self confidence determining subsystem for (a) computing a list of self confidence values, for every stored patch, (b) computing relative self confidence values, and (c) thereafter using the relative self confidence values to determine a size of a best subset of each patch to serve as a cluster candidate;</u> | <u>**NEWLY ADDED CLAIM LIMITATION**</u> |
| a cluster estimation subsystem for generating cluster data of said document-keyword vectors using said similarities of patches wherein the cluster estimation subsystem selects said patches depending on ~~inner-patch~~ <u>intra-patch</u> confidence values to represent clusters of said document keyword vectors, estimate the sizes of said patches, and generate cluster data of document keyword vectors using similarities of the patches<u>; and</u> | Tang, Column 11, lines 6-18:<br>To solve this problem, a set of grid points g.sub.11 =(1.5,1.5), g.sub.12 =(1.5,2.5), . . . with a radius of 0.5 are first chosen for searching. These grid points may be not part of the metric space E. Applying the "triangle inequality" rule of FIG. 3, query "q" 420 may be compared with all of the grid points such that nonrelevant neighborhoods are eliminated from the search and to produce a result set containing only relevant grids. The result set shows that search query "q" 420 is a subset of grids: g.sub.11, g.sub.12, g.sub.21, g.sub.22 of metric space E. As such, and as shown in the example, the search is reduced from comparing the search query "q" with all shown neighborhoods (grids) to comparing "q" with only four grids, a significant increase in efficiency.<br><br>Tang, Column 4, line 55 – column 5, line 17:<br><br>For example, for a given query point q, inexact matches to q in a given data set can be found. In mathematical terms, the search aims to find all points p in the data set such that those points p satisfy d(q,p).ltoreq..epsilon., where d( . , . ) is the distance function in the metric space, and .epsilon. is the size of interested neighborhood. When a search started, a comparison is performed among all of the grid points of at the first level of the created Bgrid tree. At each level, many subtrees are totally eliminated for further search by applying the triangular inequality rule. For example, suppose the grid points are g.sub.ij, the comparison search for the desired data elements to satisfy the search string q is only to be further carried out within those grids where d(q,g.sub.ij)<.epsilon..+.delta..sub.ij, where .delta..sub.ij is the chosen grid size. The systems and methods of the present invention perform these calculations to produce result set for communication to the participating user.<br><br>In an alternative implementation, the systems and methods of the present invention are used to calculate local distances for submitted search queries. For example, for a given query q, the search aims to find most of points p where d(p, q)<.epsilon., |

| Claim 1 As Amended | References |
|---|---|
| | whereas the missed points p are likely close to the boundary of d(p,q)=.epsilon.. In this implementation, current search algorithms, such as, BLAST and FASTA are used to create a local multigrid tree (or a local Bgrid tree) having local distances. Employing the same steps above, the local multigrid tree (or local Bgrid) tree is analyzed to find data elements for the submitted search query. Since local distances are used to create the local multigrid tree (or the local Bgrid search tree), the result set will contain most of the desired hits for a submitted search query.<br><br>Tang, column 10, line 61, to column 11, line 27:<br>As shown in FIG. 4, metric space "E" 400 can be divided into many small grids 405, 410, 415, etc., each containing a grid point, 405a, 410a, 415a, etc, respectively. FIG. 4 shows an example of a multigrid in a 2dimensional point set with L.sub.1 distance and a corresponding search performed on the grid. For example, consider a set of points E, all which are located in a 2dimensional area of [1,5][1,4]. The L.sub.1 distance may be defined as follows, given p.sub.1 =(x.sub.1,y.sub.1), p.sub.2 =(x.sub.2,y.sub.2), d(p.sub.1,p.sub.2)=max(.vertline.x.sub.1 x.sub.2.vertline.,.vertline.y.sub.1 y.sub.2.vertline.). Using this calculated distance, an exemplary search may be performed to answer the question: given a query point q=(2.2, 1.8), find out all points p within the area that satisfy d(q,p)<0.3.<br><br>To solve this problem, a set of grid points g.sub.11 =(1.5,1.5), g.sub.12 =(1.5,2.5), . . . with a radius of 0.5 are first chosen for searching. These grid points may be not part of the metric space E. Applying the "triangle inequality" rule of FIG. 3, query "q" 420 may be compared with all of the grid points such that nonrelevant neighborhoods are eliminated from the search and to produce a result set containing only relevant grids. The result set shows that search query "q" 420 is a subset of grids: g.sub.11, g.sub.12, g.sub.21, g.sub.22 of metric space E. As such, and as shown in the example, the search is reduced from comparing the search query "q" with all shown neighborhoods (grids) to comparing "q" with only four grids, a significant increase in efficiency.<br><br>The grid concept can be extended one more step such that there are multiple levels of grids. Each grid at a fixed level can be subdivided into smaller grids with smaller radius (i.e. a smaller neighborhood). Those smaller grids become children, and the original grid with its gridpoint is the parent. In this way, multiple levels of grids can be linked via parentchild relationships. As illustrated in FIG. 5, the multilayered grid structure when assembled forms a grid search tree.<br><br>Tang: Column 13, line 61 to column 14, line 16:<br><br>The process of building the Bgrid tree of FIG. 6A is described by the flow diagram of FIG. 5. This method is a slightly modified approach compared with the Btree definitions described by R. Bayer and E. McCreight, "Organization and Maintenance of Large Ordered Indexes," Acta Informatica. 1:173189 (1970), which is herein incorporated by reference. The Bgrid tree of FIG. 6A maintains each subtree within its parent grid; whereas in the conventional Btree definition, the subtree is either to the left or right of its parent node. This slight difference makes the Bgrid tree concept uniform to all space dimensions in a metric space.<br><br>FIG. 8B shows an exemplary Bgrid tree of order 4 in a 2dimensioanl metric space. Similar to the Bgrid tree of FIG. 8A, each grid is defined by a grid point 850, a radius 855, and some descriptions 860. A shown, there are four grids g.sub.1 (865), g.sub.2 (870), g.sub.3 (875), and g.sub.4 (880). The neighborhoods defined by the grid points and the radii may be |

| Claim 1 As Amended | References |
|---|---|
| | overlapping, but as the descriptions indicate, these neighborhoods exist separate and apart. Thus, the grids at the same level have no overlapping regions (points). The descriptions are provided to assign the points in overlapping regions to one of the grids.<br><br>Kanno, column 21, line 64 – column 22, line 54:<br><br>(Constitution of Similar Vector Searching Apparatus)<br><br>FIG. 4 is a block diagram showing the whole constitution of the similar vector searching apparatus according to claims 10, 11, 13, 23, 24, 26 of the present invention. In FIG. 4, a vector index 401 is prepared by the vector index preparing apparatus of the aforementioned first embodiment, and is a vector index prepared from the vector database which stores 200,000 pieces of vector data constituted of two items of: the 296-dimensional real vector prepared from the newspaper article full text database of 200,000 collected newspaper articles and indicating the characteristic of each newspaper article; and the identification number of 1 to 200,000 for uniquely identifying each article and which has the content as shown in FIGS. 12A and 12B.<br><br>In order to perform the similarity search on the newspaper article full text database, search condition input means 402 inputs the identification number of any article in the newspaper article full text database, and the similarity lower limit value and maximum obtained pieces number of 0 to 100 indicating the similarity search range, searches the vector index 401 with the identification number to obtain the vector of the corresponding article as the query vector Q from the inputted identification number, and obtains a square distance from the similarity lower limit value, that is, obtains a square distance upper limit value .alpha..sup.2 as the upper limit value of the squared distance.<br><br>Partial query condition calculation means 403 calculates a partial square distance upper limit value f.sup.2 as the upper limit value of the square distance of 37 types of 8-dimensional partial query vectors q and the partial vector corresponding to q by f.sup.2=.alpha..sup.2\|q\|.sup.2/\|Q\|.sup.2 with respect to partial spaces of 0 to 36 for the query vector Q obtained by the search condition input means 402.<br><br>Search object range generation means 404 enumerates all sets (d, c, [r.sub.1, r.sub.2]) of the region number d for specifying a region including a partial vector whose partial square distance with the partial query vector q is possibly smaller than the partial square distance upper limit value f.sup.2, declination division number c, and norm division range [r.sub.1, r.sub.2] from the partial query vector q and partial square distance upper limit value f.sup.2 obtained by the partial query condition calculation means 403 for the partial space b and the norm division table and declination division table in the vector index 401.<br><br>Index search means 405 calculates the search condition K for the vector index 401 from (d, c, [r.sub.1, r.sub.2]) generated by the search object range generation means 404 for each partial space b similarly as calculation of the key during the vector index preparation as follows. K=[k.sub.min, k.sub.max] k.sub.min=b7617440+d1024+c256+r.sub.1 k.sub.max=b7617440+d1024+c256+r.sub.2 The index search means then searches the range of the vector index 401 with the search condition K and obtains all sets (i, v) of the partial vector v and identification number i having the key to match the search condition<br><br>Kanno, Column 23, lines 6-29:<br>Similarity search result determination means 408 searches the |

| Claim 1 As Amended | References |
|---|---|
| | vector index 401 with the identification number i in order from a positive large square distance difference upper limit value S[i] in the element S[i] of the square distance difference table 407 to obtain the corresponding vector V, calculates a square distance difference value .alpha..sup.2-\|V-Q\|.sup.2 by subtracting the square distance \|V-Q\|.sup.2 of V and query vector Q calculated by the search condition input means 402 from the square distance upper limit value .alpha..sup.2 calculated by the search condition input means 402, and replaces S[i] with the square distance difference value .alpha..sup.2-\|V-Q\|.sup.2. The number of articles which have the square distance difference values larger than the maximum value of the partial square distance difference accumulated value of the article having the square distance difference value not calculated and whose square distance difference value is calculated reaches L or more. At this time, or at the time the square distance difference values of all the articles having positive partial square distance difference accumulated values are calculated, for L result candidates at maxim (i, S[i]) having positive and large square distance difference values, a set (i, sqrt(.alpha..sup.2-S[i])) of the identification number i and distance sqrt(.alpha..sup.2-S[i]) is outputted as a search result to search result output means.<br>Gilmour, Column 16, lines 32-61:<br>At step 192, an initial confidence level values for the term is then determined based on the summed adjusted counts and the term weight, as determined above with reference to the weight table 210 shown in FIG. 11. To this end, FIG. 13 illustrates a confidence level table 230, which includes various initial confidence level values for various summed adjusted count/weight value combinations that may have been determined for a term. For example, a term having a summed adjusted count of 0.125, and a weight value of 300, may be allocated an initial confidence level value of 11.5. Following the determination of an initial confidence level value, confidence level values for various terms may be grouped into "classes", which still retain cardinal meaning, but which standardize the confidence levels into a finite number of "confidence bands". FIG. 14 illustrates a modified table 240, derived from the confidence level table 230, wherein the initial confidence levels assigned are either rounded up or rounded down to certain values. By grouping into classes by rounding, applications (like e-mail addressing), can make use of the classes without specific knowledge/dependence on the numerical values. These can then be tuned without impact to the applications. The modified confidence level values included within the table 240 may have significance in a number of applications. For example, users may request that terms with a confidence level of greater than 1000 automatically be published in a "public" portion of their user knowledge profile. Further, e-mail addressees for a particular e-mail may be suggested based on a match between a term in the e-mail and a term within the user knowledge profile having a confidence level value of greater than, merely for example, 600. |
| <u>a redundant cluster elimination subsystem for using the inner patch confidence values to eliminate redundant cluster candidates.</u> | **NEWLY ADDED CLAIM LIMITATION** |

1. Applicant's claims contain the claim limitation

a neighborhood patch generation subsystem for generating <u>groups of nodes having similarities</u> as determined using a search structure, said neighborhood patch generation subsystem including a subsystem for generating a spatial approximation sample hierarchy structure upon said document-keyword vectors and <u>a patch defining subsystem for creating patch relationships among said nodes with respect to a metric distance between nodes</u>

The cited portions of Tang do not contain any recitation of either "groups of nodes having similarities" or "a patch defining subsystem for creating patch relationships among said nodes with respect to a metric distance between nodes." To the contrary, Tang describes a grid system. There is no recitation of "grids" anywhere in Applicant's invention. The algorithms are seen to be significantly different.

2. Applicant's claims contain the claim limitation

    a query vector generation subsystem accepting search conditions and query keywords, generating a corresponding query vector, and storing the generated query vector

By way of contrast, Tang discloses (Tang, Column 4, line 55 to column 5, line 3) that "... for a given query point q, inexact matches to q in a given data set can be found. In mathematical terms, the search aims to find all points p in the data set such that those points p satisfy d(q,p).≤ epsilon, where d( . , . ) is the distance function in the metric space, and epsilon is the size of interested neighborhood. When a search started, a comparison is performed among all of the grid points of at the first level of the created B-grid tree. At each level, many subtrees are totally eliminated for further search by applying the triangular inequality rule. ... as well as a "distance function" (Tang, Column 7, lines 32-41).

Next Tang describes searching multigrid trees (Tang, Column 11, line 64 to column 12, line 32)

Kanno, Column 15, lines 34-40, describes matrix operations on query vectors, however Applicants neither recites nor claims matrix operations, such as "inner products."

Applicant's claims next recite "an intra-patch confidence and intrapath confidence determination subsystem for every element of the database, the spatial approximation sample hierarchy structure computing a neighborhood patch consisting of a list of those database elements most similar to it for computing intra-patch confidence values between patches and interpath confidence values" to which there is no corresponding teaching in Tang, Kanno, or Gilmour.

3. Next Applicant claims “a cluster estimation subsystem for generating cluster data of said document-keyword vectors using said similarities of patches wherein the cluster estimation subsystem selects said patches depending on intra-patch confidence values to represent clusters of said document keyword vectors, estimate the sizes of said patches, and generate cluster data of document keyword vectors using similarities of the patches”

This is neither taught nor suggested by Tang’s disclosures of choosing grid points for searching (Tang, Column 11, lines 6-18), and applying the "triangle inequality" rule with elimination of non-relevant neighborhoods to produce a result set containing only relevant grids, or by dividing the sample space into many small grids, with each containing a grid point, and using this calculated distance to perform an exemplary search to answer the question:”given a query point q=(2.2, 1.8), find out all points p within the area that satisfy d(q,p)<0.3.”

Kanno, column 21, line 64 – column 22, line 54 describes an alternative vector searching apparatus.

Gilmour, Column 16, lines 32-61, describes obtaining initial confidence level values for the term is based on the summed adjusted counts and the term weight, as determined above with reference to a weight table.

4. The claim limitation “a redundant cluster elimination subsystem for using the inner patch confidence values to eliminate redundant cluster candidates” is newly added.

The art of record neither teaches nor suggests applicants’ claimed invention.

## Conclusion

Based on the above discussion, it is respectfully submitted that the pending claims describe an invention that is properly allowable to the Applicants.

If any issues remain unresolved despite the present amendment, the Examiner is requested to telephone Applicants' Attorney at the telephone number shown below to arrange for a telephonic interview before issuing another Office Action.

Applicants would like to take this opportunity to thank the Examiner for a thorough and competent examination and for courtesies extended to Applicants' Attorney.

| **Certificate of Electronic Deposit** | Respectfully Submitted |
|---|---|
| I hereby certify that this paper (along with any paper referred to as being attached or enclosed) is being deposited with the United States Patent and Trademark Office on the date shown below | /s/Richard M. Goldman/s/ |
| Date of deposit: 10-14-2007<br>Person mailing paper: Richard M. Goldman<br>Signature: /s/Richard M. Goldman/s/ | Richard M. Goldman, Reg. # 25,585<br>371 Elan Village Lane, Suite 208<br>San Jose, CA 95134<br>Voice: 408-324-0716<br>Fax: 408-324-0672<br>E-mail: goldmanptn@aol.com |